Amendments to the Claims

The following listing of claims will replace all prior versions, and listings, of claims in the ancestor application.

1. (Previously Presented) A method for performing concurrent mark-sweep garbage collection, comprising:

receiving an application, the application including a software application to run concurrently with mark-sweep garbage collection;

executing the application in at least one thread, the application using space in a heap while being executed;

determining if available space in the heap falls below a threshold;

performing mark-sweep garbage collection, concurrently while executing the application, in a heap block of the heap using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer in the heap block, if the available space falls below the threshold; and otherwise,

continuing executing the application and monitoring if the available space in the heap falls below the threshold, until the execution of the application is complete;

wherein performing mark-sweep garbage collection includes invoking at least one garbage collection thread to trace and mark live objects in the heap block concurrently while executing the application; and wherein the mark bit vector pointer initially points to the first bit vector, the sweep bit vector pointer initially points to the second bit vector, and the first bit vector and the second bit vector are toggled with each other after a marking phase of the mark-sweep garbage collection completes.

2. (Original) The method of claim 1, wherein the heap comprises at least one heap block.

3. (Original) The method of claim 1, further comprising initializing a concurrent mark-sweep garbage collector.

4. (Original) The method of claim 3, wherein initializing the concurrent mark-sweep garbage collector comprises setting each bit in the first bit vector and the second bit vector to 0, and pointing the mark bit vector pointer to the first bit vector and the sweep bit vector pointer to the second bit vector.

5. (Previously Presented) The method of claim 1, wherein performing mark-sweep garbage collection further comprises:

reclaiming storage space occupied by objects other than the live objects in the block concurrently while tracing the live objects in the block and executing the application.

6. (Original) The method of claim 5, wherein tracing the live objects in the heap block comprises parallel marking the live objects by at least one garbage collection thread.

7. (Original) The method of claim 6, wherein parallel marking the live objects comprises setting bits corresponding to starting addresses of the live objects in a bit vector pointed to by the mark bit vector pointer to 1, by the at least one garbage collection thread.

8. (Original) The method of claim 5, wherein reclaiming the storage space occupied by objects other than the live objects in the heap block comprises sweeping the heap block to make the said storage space allocable by using a bit vector pointed to by the sweep bit vector pointer.

9. (Cancelled)

10. (Original) The method of claim 8, further comprising setting the bit vector back to 0 after completing sweeping the heap block.

11. (Original) The method of claim 1, further comprising performing another cycle of concurrent mark-sweep garbage collection when available space in the heap falls below the threshold again.

12. (Currently Amended) A method for automatically collecting garbage objects, comprising:

receiving a first code, the first code being neutral to a computing platform;

compiling the first code into a second code, the second code being native to the computing platform;

executing the second code in at least one thread by the computing platform; and

automatically performing mark-sweep garbage collection concurrently with ~~the executing~~ executing the second code, to ensure there is storage space available for executing the second code;

wherein automatically performing mark-sweep garbage collection detects if available space in a heap falls below a threshold and invokes the concurrent mark-sweep garbage collection when the available space falls below the threshold by using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer for a heap, the mark bit vector pointer initially pointing to the first bit vector, the sweep bit vector pointer initially pointing to the second bit vector, and the first bit vector and the second bit vector being toggled with each other after a marking phase of the mark-sweep garbage collection completes.

13. (Cancelled)

14. (Previously Presented) The method of claim 12, wherein the heap comprises at least one heap block.

15. (Cancelled)

16. (Previously Presented) A system for concurrent mark-sweep garbage collection, comprising:

a root set enumeration mechanism to enumerate references to live objects in a heap;

a live object tracing mechanism to parallel trace live objects in a heap block and mark the live objects in a first bit vector pointed to by a mark bit vector pointer in the heap block, concurrently with execution of an application, the application using space in the heap while being executed, the live object tracing mechanism including a live object search mechanism to parallel search live objects in the heap block by at least one garbage collection thread, and a live object marking mechanism to parallel mark the live objects in a bit vector stored in the heap block by the at least one garbage collection thread;

a garbage sweeping mechanism to sweep storage space occupied by garbage objects to make the storage space allocable using a second bit vector pointed to by a sweep bit vector pointer in the heap block, concurrently with the execution of the application and live object marking; and

a bit vector toggling mechanism to toggle the first bit vector pointed to by the mark bit vector pointer and the second bit vector pointed to by the sweep bit vector pointer in the heap block.

17. (Cancelled)

18. (Previously Presented) The system of claim 16, wherein the live object tracing mechanism further comprises:

a live object scanning mechanism to parallel scan any objects reachable from the live objects in the heap; and

a conflict prevention mechanism to prevent more than one garbage collection thread from marking the same object.

19. (Previously Presented) A system for managing memory usage during run-time, comprising:

a just-in-time compiler to compile an application into a code native to an underlying computing platform;

a virtual machine to execute the application, the application using space in a heap while being executed; and

a garbage collector to trace live objects, mark the live objects in a first bit vector pointed to by a mark bit vector pointer in a heap block of the heap, and toggle the bit first vector pointed to by the mark bit vector pointer with a second bit vector pointed to by a sweep bit vector pointer at the end of marking phase, concurrently with execution of the application, the garbage collector including a live object marking mechanism to parallel mark the live objects in the first bit vector pointed to by the mark bit vector in the heap block of the heap, and a bit vector toggling mechanism to toggle the first bit vector pointed to by the mark bit vector pointer and the second bit vector pointed to by the sweep bit vector pointer.

20. (Original) The system of claim 19, further comprising a garbage sweeping mechanism to sweep storage space occupied by garbage objects to make the storage space allocable using a bit vector pointed to by the sweep bit vector pointer, concurrently with the execution of the application and live objects marking.

21-24. (Cancelled)

25. (Previously Presented) An article comprising: a machine accessible medium having content stored thereon, wherein when the content is accessed by a processor, the content provides for performing concurrent mark-sweep garbage collection by:

receiving an application, the application including a software application to run concurrently with mark-sweep garbage collection;

executing the application in at least one thread, the application using space in a heap while being executed;

determining if available space in the heap_falls below a threshold;

performing mark-sweep garbage collection, concurrently while executing the application, in a heap block of the heap using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer in the heap block, if the available space falls below the threshold; and otherwise,

continuing executing the application and monitoring if the available space in the heap falls below the threshold, until the execution of the application is complete;

wherein performing mark-sweep garbage collection includes invoking at least one garbage collection thread to trace and mark live objects in the heap block concurrently while executing the application; and wherein the mark bit vector pointer initially points to the first bit vector, the sweep bit vector pointer initially points to the second bit vector, and the first bit vector and the second bit vector are toggled with each other after a marking phase of the mark-sweep garbage collection completes.

26. (Original) The article of claim 25, wherein the heap comprises at least one heap block.

27. (Original) The article of claim 25, further comprising content for initializing a concurrent mark-sweep garbage collector.

28. (Original) The article of claim 27, wherein content for initializing the concurrent mark-sweep garbage collector comprises content for setting each bit in the first bit vector and the second bit vector to 0, and pointing the mark bit vector pointer to the first bit vector and the sweep bit vector pointer to the second bit vector.

29. (Previously Presented) The article of claim 25, wherein content for performing mark-sweep garbage collection further comprises content for:

reclaiming storage space occupied by objects other than the live objects in the block concurrently while tracing the live objects in the block and executing the application.

30. (Original) The article of claim 29, wherein content for tracing the live objects in the heap block comprises content for parallel marking the live objects by at least one garbage collection thread.

31. (Original) The article of claim 30, wherein content parallel marking the live objects comprises content for setting bits corresponding to starting addresses of the live objects in a bit vector pointed to by the mark bit vector pointer to 1, by the at least one garbage collection thread.

32. (Original) The article of claim 29, wherein content for reclaiming the storage space occupied by objects other than the live objects in the heap block comprises content sweeping the heap block to make the said storage space allocable by using a bit vector pointed to by the sweep bit vector pointer.

33. (Cancelled)

34. (Original) The article of claim 32, further comprising content for setting the bit vector back to 0 after completing sweeping the heap block.

35. (Original) The article of claim 25, further comprising content performing another cycle of concurrent mark-sweep garbage collection when available space in the heap falls below the threshold again.

36. (Currently Amended) An article comprising: a machine accessible medium having content stored thereon, wherein when the content is accessed by a processor, the content provides for automatically collecting garbage objects by:

receiving a first code, the first code being neutral to a computing platform;

compiling the first code into a second code, the second code being native to the computing platform;

executing the second code in at least one thread by the computing platform; and

automatically performing mark-sweep garbage collection concurrently with ~~the executing~~ executing the second code, to ensure there is storage space available for executing the second code;

wherein automatically performing mark-sweep garbage collection detects if available space in a heap falls below a threshold and invokes the concurrent mark-sweep garbage collection when the available space falls below the threshold by using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer for a heap, the mark bit vector pointer initially pointing to the first bit vector, the sweep bit vector pointer initially pointing to the second bit vector, and the first bit vector and the second bit vector being toggled with each other after a marking phase of the mark-sweep garbage collection completes.

37. (Cancelled)

38. (Previously Presented) The article of claim 36, wherein the heap comprises at least one heap block.

39. (Cancelled)